Beet Seed  
  
1. Функціональне тестування:

Що перевіряється:

- Перевірка функціональності системи.

- Перевірка того, чи виконуються вимоги та очікування замовника.

- Тестування реакції системи на вхідні дані та вивід результатів.

Коли застосовується:

- Проводиться під час розробки нового продукту або під час внесення змін до існуючого.

- Використовується для валідації функціональних вимог.

Обмеження:

- Не вивчає внутрішню структуру програми.

- Фокусується на тому, що система робить, а не на тому, як вона це робить.

Особливості:

- Використовує тест-кейси для виконання конкретних дій та перевірки результатів.

- Зазвичай, автоматизується для ефективності та повторюваності.

Нефункціональне тестування:

Що перевіряється:

- Перевірка аспектів системи, не пов'язаних із конкретними функціями (продуктивність, безпека, масштабованість).

- Тестування відповідності стандартам та регуляторним вимогам.

- Перевірка ефективності, надійності та продуктивності системи.

Коли застосовується:

- Виконується після функціонального тестування.

- Основний фокус на аспектах, що не пов'язані з конкретним функціоналом.

Обмеження:

- Зазвичай не зосереджується на конкретних функціях, але може включати елементи функціонального тестування.

Особливості:

- Включає тести продуктивності, безпеки, доступності та інші характеристики системи.

- Тестується на масштабованість та відповідність стандартам.

Тестування, пов’язане зі змінами:

Що перевіряється:

- Перевірка впливу змін на існуючий функціонал та взаємодію з іншими частинами системи.

- Тестування забезпечення, що нові функції або зміни не порушують існуючий функціонал.

Коли застосовується:

- Використовується при внесенні змін до програмного продукту або його компонентів.

- Може включати регресійне тестування для перевірки впливу на існуючий код.

Обмеження:

- Може вимагати розгортання окремого тестового середовища для перевірки змін.

Особливості:

- Включає тестування взаємодії нового функціоналу з існуючим та забезпечення сумісності.

- Регулярно використовується під час розробки та вдосконалення продукту.

Beet Sprout

2. Можливе проведення тільки функціонального тестування без перевірки нефункціональних вимог у певних випадках, але це може бути обмеженням і призвести до недоліків у продукті. Давайте розглянемо обидві сторони цього питання.

Можливість проведення тільки функціонального тестування може бути виправданою у випадках, коли:

* Обмежений обсяг часу і ресурсів: У випадках, коли обсяг часу або ресурсів обмежений, команді може бути важко або неможливо провести повноцінне тестування всіх аспектів продукту. Тоді фокус може бути зосереджений на функціональності, яка є основною для користувачів.
* Вимоги замовника: Якщо замовник вклав акцент на функціональність та не вважає нефункціональні вимоги критичними. Якщо продукт визначається як прототип або етап мінімальної життєздатності, і його функціональність важливіша для випробувань на ринку.

Незважаючи на це, проведення тільки функціонального тестування може призвести до наступних недоліків:

* Недоліки з точки зору продуктивності: Без перевірки продуктивності можуть виникнути проблеми з відгуком, швидкістю та роботою системи під великим навантаженням.
* Безпека: Нефункціональні вимоги, пов'язані з безпекою, такі як захист від атак, можуть залишитися невипробуваними, що створює ризик для системи.
* Доступність: Нефункціональні аспекти, такі як доступність для людей з обмеженими можливостями, можуть бути проігноровані, що вплине на аудиторію користувачів.
* Ефективність: Функціональне тестування може не виявити проблеми з ефективністю, такі як неоптимальне використання ресурсів.

3. Smoke тестування (димове тестування) є однією з форм функціонального тестування програмного забезпечення, призначеною для перевірки основних функцій системи без деталей. Основна мета smoke тестування - визначити, чи може програма працювати базовим чином перед тим, як переходити до більш докладних тестів.

Ось кілька причин, чому smoke тестування є важливим етапом у розробці програмного забезпечення:

* Базова стабільність:

- Smoke тест допомагає визначити, чи існують серйозні проблеми, які можуть перешкоджати базовому функціоналу програми. Це особливо важливо на початкових етапах розробки.

* Визначення проблем на ранніх етапах: Димове тестування дозволяє виявляти серйозні проблеми якнайраніше, що зменшує витрати на виправлення помилок на пізніших етапах розробки.
* Оцінка якості основних функцій: Воно дозволяє оцінити якість основних функцій продукту, що дозволяє розробникам та тестувальникам швидше зрозуміти, чи відповідає продукт основним вимогам.
* Економія часу та ресурсів: Smoke тестування виконується швидко і дозволяє виявити критичні проблеми, не затрачаючи багато часу на детальне тестування.